Clustering network data through effective use of eigensolvers and hypergraph models

Alicia Klinvex, Michael Wolf, and Daniel Dunlavy

Sandia National Laboratories

Exceptional service in the national interest

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

CCR Center for Computing Research

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2016-2317 C

Motivating problem: Community detection

- Determine groupings of data objects given sets of relationships amongst those objects
- Relationships may be represented in a graph or hypergraph
  - Graphs represent pairwise relationships
  - Hypergraphs represent relationships among groups of things
- Applications
  - Finding emerging research trends from documents (Jung et al., 2014)
  - Clustering categorical data (Gibson et al., 2000)
  - Image segmentation (Agarwal et al., 2005)
  - Metabolic networks (Guimera et al., 2004)

Outline

- Introduction to hypergraphs
- Description of spectral clustering algorithm
- Exploration of eigenvalue problems occurring in spectral clustering
- Spectral clustering results

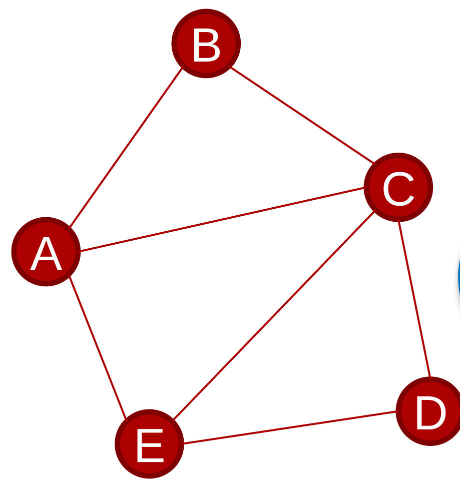
- Explore the usage of hypergraphs to model relational data
- Understand how to effectively use eigensolvers in spectral analysis of this data

What is a hypergraph?

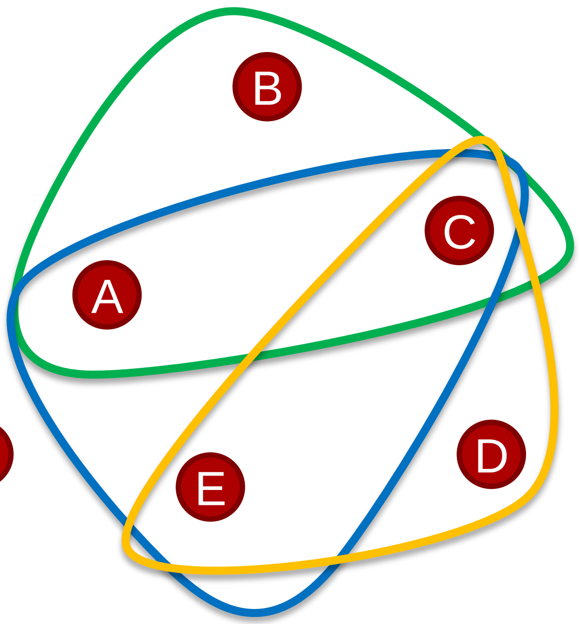
Docs

Authors	1	2	3
A	X		X
B	X		
C	X	X	X
D		X	
E		X	X

Graph



Hypergraph



- Generalization of graph
  - Hyperedges represent multiway relationships between vertices
  - A hyperedge is a set of vertices of arbitrary size
  - Hyperedges can connect more than 2 vertices

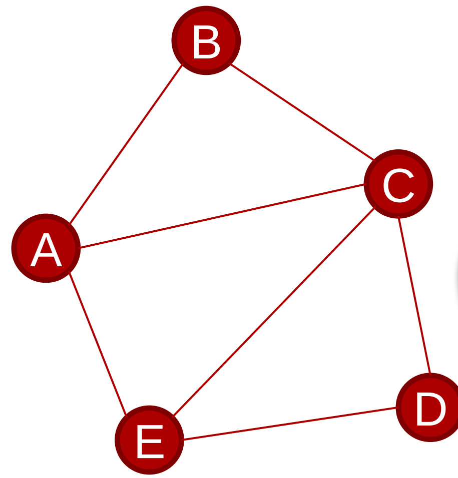
What is a hypergraph?

Docs

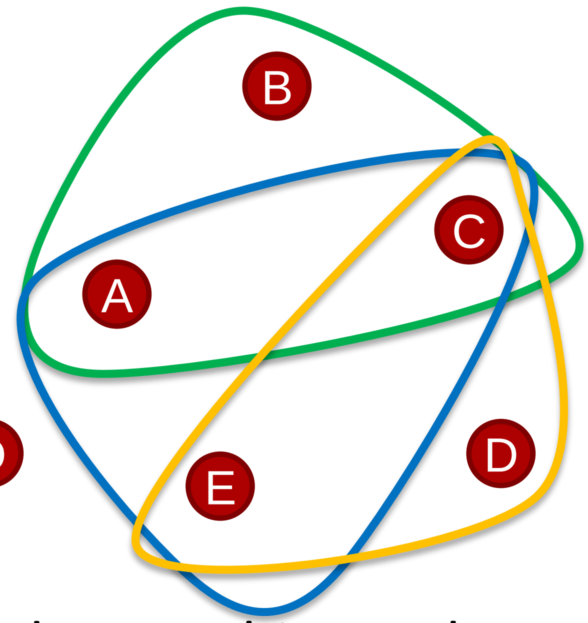
Authors	1	2	3
A	1		1
B	1		
C	1	1	1
D		1	
E		1	1

Hypergraph incidence matrix

Graph



Hypergraph



- Multiway relationships can be represented nonambiguously
  - Did A, B, and C write a paper together?
- Relational data is hypergraph incidence matrix
  - One way to represent a hypergraph as a graph: clique expansion

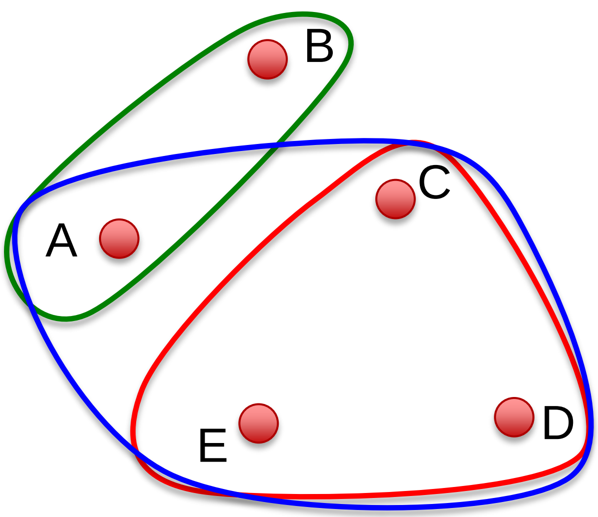
Hypergraph clique expansion

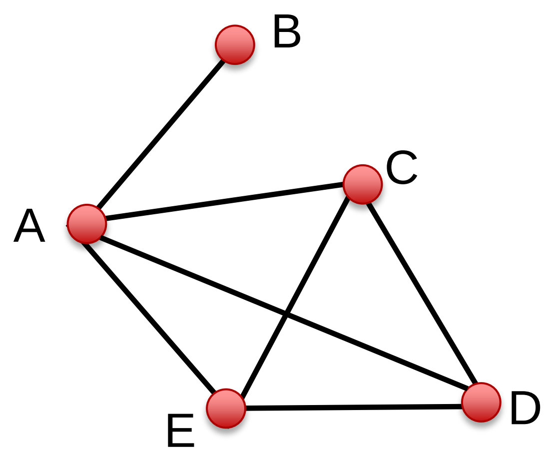
Hyperedges

Vertices	1	2	3
A	1		1
B	1		
C		1	1
D		1	1
E		1	1

Graph Edges

	1	2	3	4	5	6	7	8	9	10
A	X				X	X	X			
B	X									
C		X	X		X			X	X	
D		X		X		X		X		X
E			X	X			X		X	X





$$|E_g| = \sum_{e_h \in E_h} \binom{d(e_h)}{2}$$

$d(e_h)$:
hyperedge cardinality

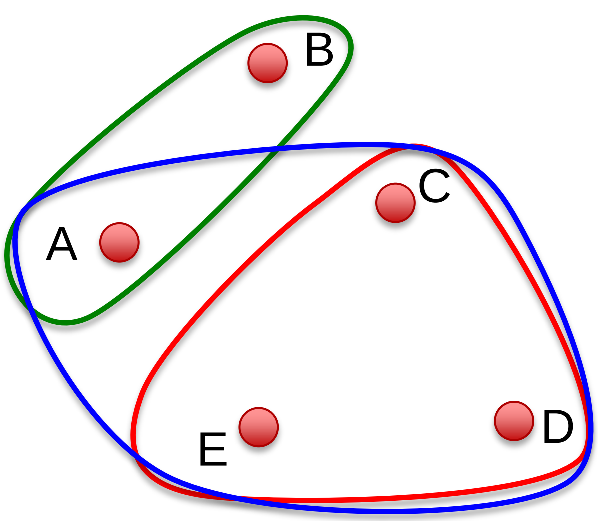
Weighted hypergraph clique expansion

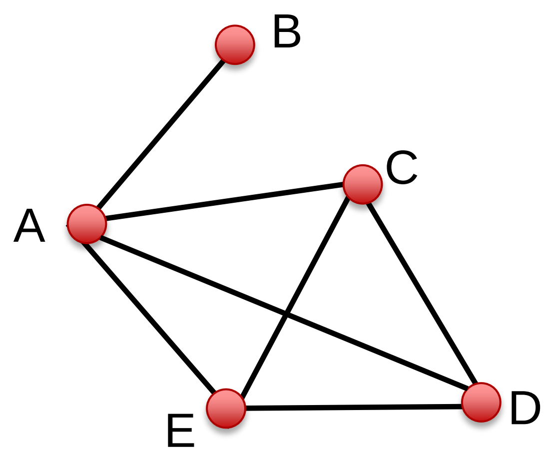
Hyperedges (rows: Vertices)

	1	2	3
A	1		1
B	1		
C		1	1
D		1	1
E		1	1

Graph Edges

	1	2	3	4	5	6	7	8	9	10
A	1				⅓	⅓	⅓			
B	1									
C		½	½		⅓			⅓	⅓	
D		½		½		⅓		⅓		⅓
E			½	½			⅓		⅓	⅓





$$w(e_g) = \frac{1}{d(e_h) - 1}$$

$d(e_h)$:
hyperedge cardinality

Computational advantages of hypergraphs

1		1
1		
	1	1
	1	1
	1	1

Hypergraph incidence matrix

1				⅓	⅓	⅓			
1									
	½	½		⅓			⅓	⅓	
	½		½		⅓		⅓		⅓
		½	½			⅓		⅓	⅓

Graph Incidence matrix

- Hypergraphs require significantly **less storage space** than graphs generated using clique expansion

$$|E_g| = \sum_{e_h \in E_h} \binom{d(e_h)}{2}$$

- Hypergraphs require **fewer operations** for a matrix-vector multiplication

How do we detect communities in graphs and hypergraphs?

- One way: spectral clustering (Ng, et al., 2002)
  - Compute the smallest eigenpairs of the normalized graph or hypergraph Laplacian*

$$L_H = I - {D_v}^{-1/2} H_h {D_e}^{-1} {H_h}^T {D_v}^{-1/2} \in \mathbb{R}^{n \times n}$$

$$L_G = I - {D_v}^{-1/2} \left(H_g {H_g}^T - D_v \right) {D_v}^{-1/2} \in \mathbb{R}^{n \times n}$$

    - **Laplacian is never explicitly formed**

$$H_h = \begin{vmatrix} 1 & & 1 \\ 1 & & \\ & 1 & 1 \\ & 1 & 1 \\ & 1 & 1 \end{vmatrix} \qquad H_g = \begin{vmatrix} 1 & & & & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & & & \\ 1 & & & & & & & & & \\ & \frac{1}{2} & \frac{1}{2} & & \frac{1}{3} & & & \frac{1}{3} & \frac{1}{3} & \\ & \frac{1}{2} & & \frac{1}{2} & & \frac{1}{3} & & \frac{1}{3} & & \frac{1}{3} \\ & & \frac{1}{2} & \frac{1}{2} & & & \frac{1}{3} & & \frac{1}{3} & \frac{1}{3} \end{vmatrix}$$

*We use the hypergraph Laplacian definition presented in (Zhou, et al., 2006)

How do we detect communities in graphs and hypergraphs?

- Spectral clustering (Ng, et al., 2002)
  - Compute the smallest eigenpairs of the normalized graph or hypergraph Laplacian (Zhou, et al., 2006)

$$L_G = I - D_v^{-1/2}(H_g H_g^T - D_v) D_v^{-1/2}$$

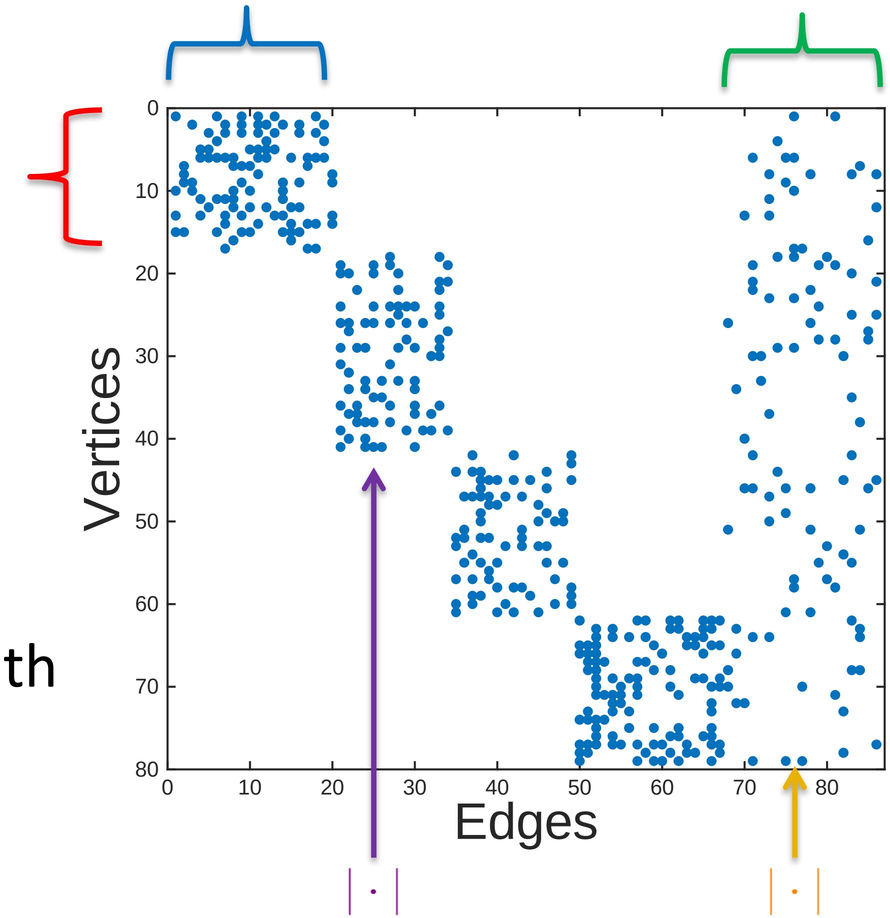
$$L_H = I - D_v^{-1/2} H_h D_e^{-1} H_h^T D_v^{-1/2}$$

  - Perform k-means clustering on those eigenvectors
    - Partition a set of observations into clusters in which each observation belongs to the cluster with the nearest mean
- Quality of our results is measured using the Jaccard index
  - T = true cluster assignments
  - P = predicted cluster assignments

$$J(T,P) = \frac{|T \cap P|}{|T \cup P|}$$

Randomly Generated Hypergraphs

- Parameters
  - Clusters
  - Nodes per cluster
  - Intra-cluster hyperedges
  - Inter-cluster hyperedges
  - Hyperedge cardinalities
    - Intra-cluster
    - Inter-cluster
- We also generate a ground truth clustering vector
- We may generate multiple instances with the same set of parameters



Incidence Matrix: H_h

EFFECTIVE USE OF EIGENSOLVERS

Experimental results

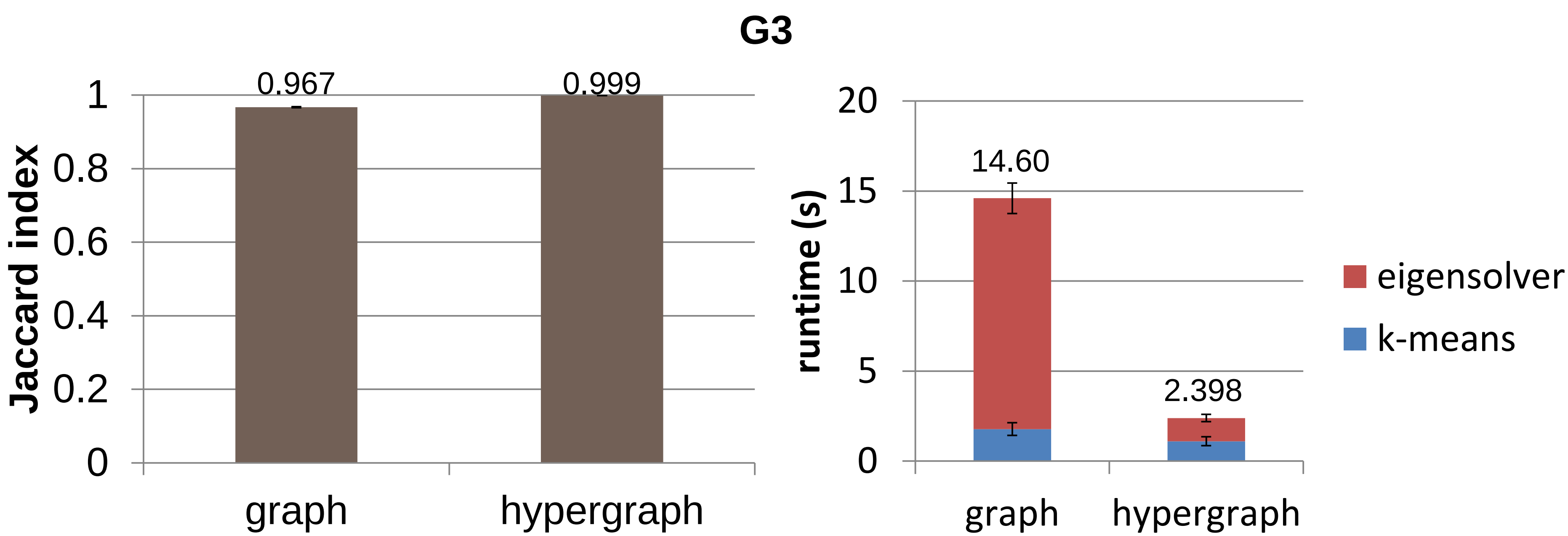
- Experiments were conducted on a 24 core machine with 128 GB of memory using 16 MPI processes
- Runtime parameters
  - 10 randomly generated hypergraphs of each type

	G1	G2	G3
Number of clusters	10	5	10
Nodes per cluster	10,000	10,000	10,000
Intra/Inter-cluster hyperedges	40,000 / 50,000	20,000 / 200,000	20,000 / 200,000
Intra/Inter-cluster h-edge cardinality	5 / 5	10 / 3	5 / 5

  - 5 k-means trials per matrix
  - Eigensolver: LOBPCG (available in Trilinos)
  - Number of computed eigenpairs: same as number of clusters*
  - Tolerance: 1e-3*

*unless otherwise stated

How do graph and hypergraph results compare?

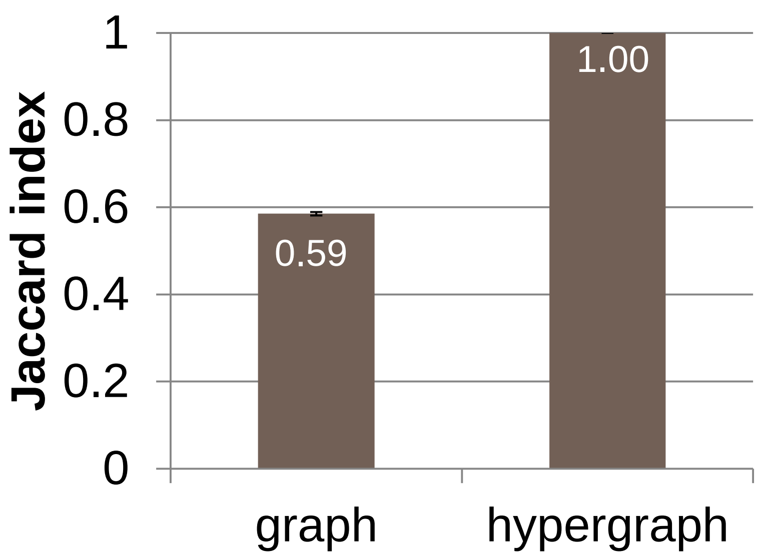


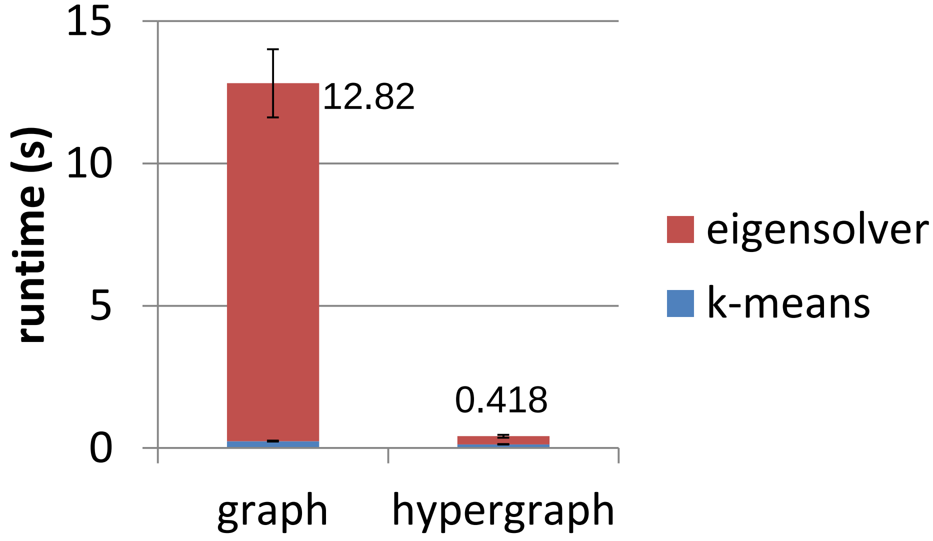
	Graph	Hypergraph
k-means iterations	79.4	28.1
LOBPCG iterations	15.6	8.9

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	5 5

How do graph and hypergraph results compare?

G2

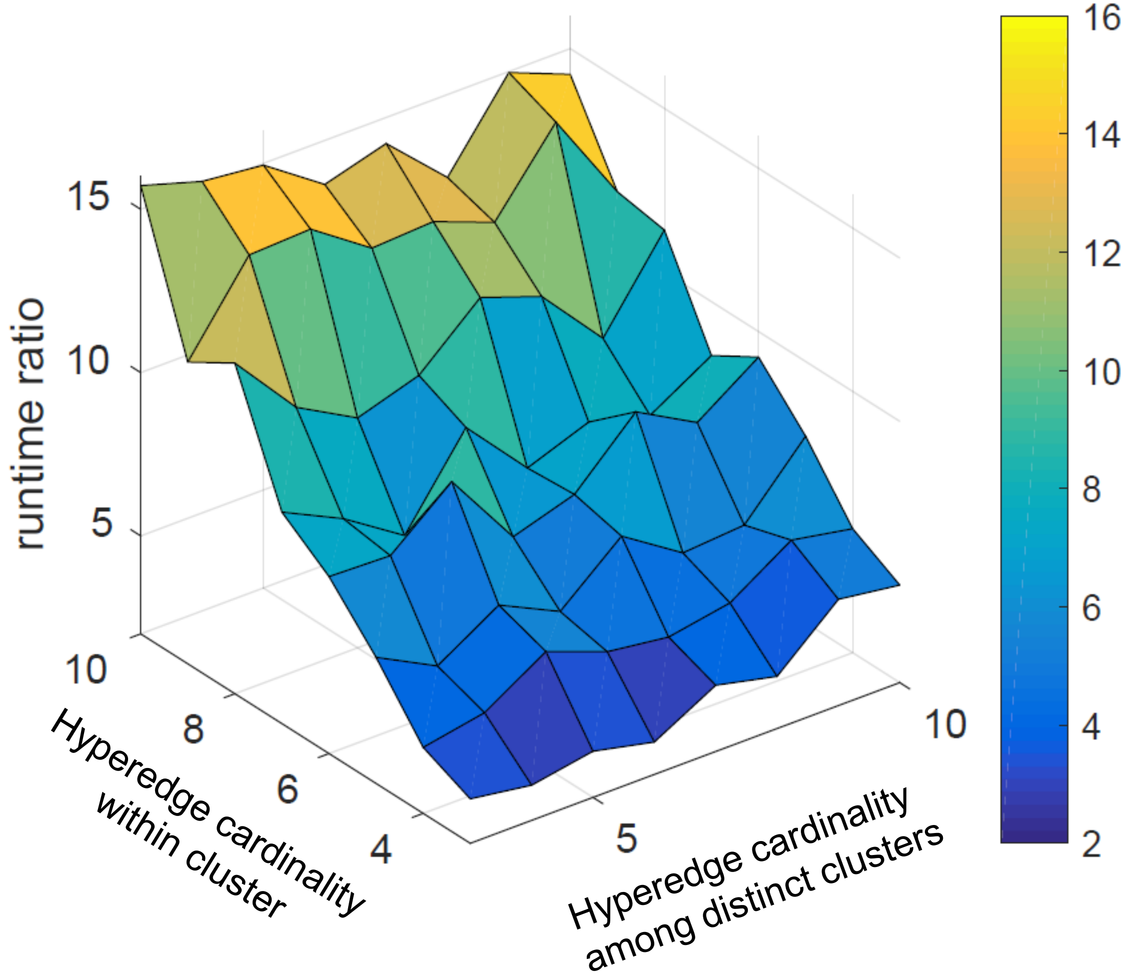




	Graph	Hypergraph
k-means iterations	56.8	5.4
LOBPCG iterations	31.1	6.5

Number of clusters	5
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	10 3

How do graph and hypergraph runtimes compare?



- **Runtime ratio**:

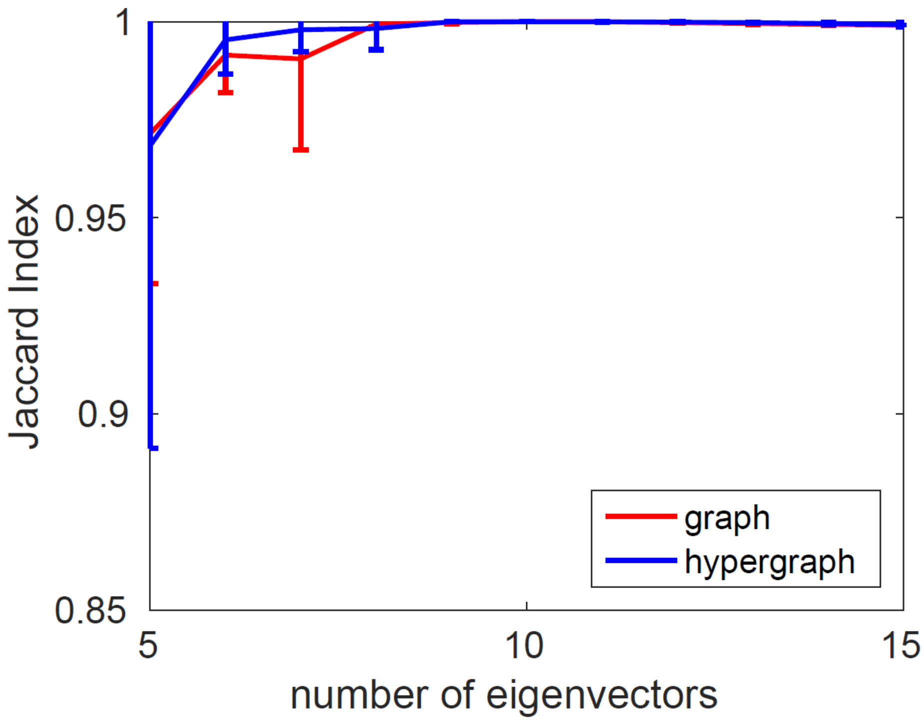
$$\frac{\text{graph runtime}}{\text{hypergraph runtime}}$$

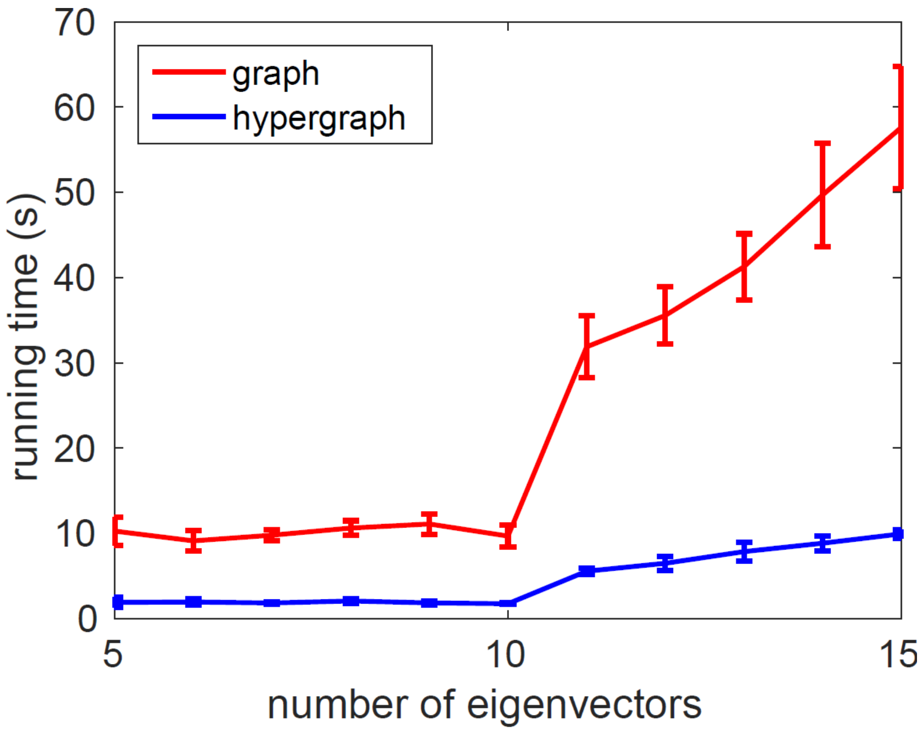
- Large numbers: bad

Number of clusters	5
nodes per cluster	10,000
Intra/Inter-cluster hyperedges	40,000 50,000

Jaccard index was always 1

How many eigenvectors should we calculate?

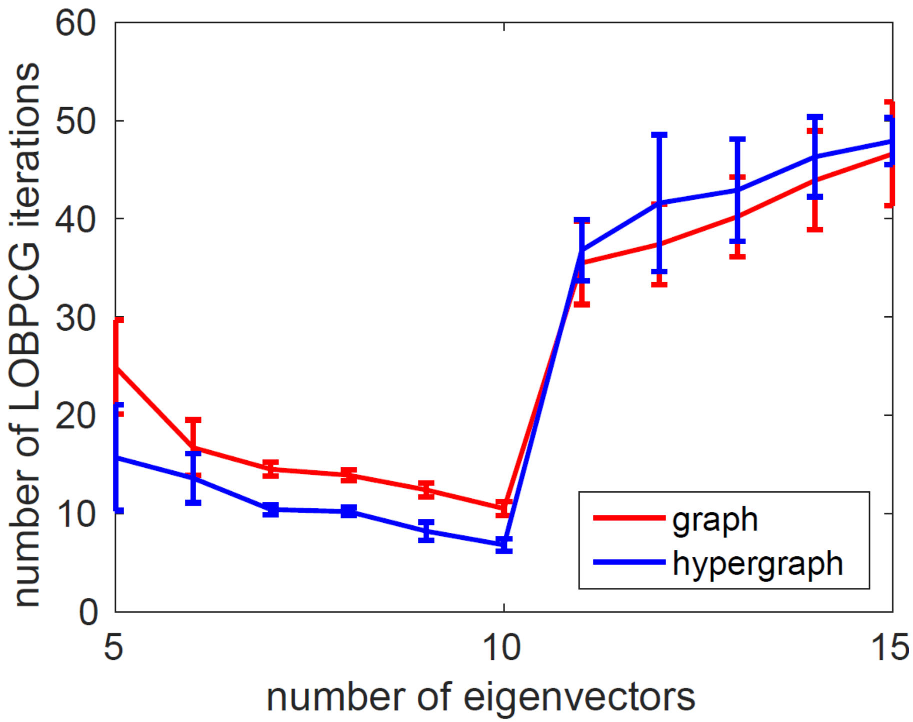




Less noisy data: G1

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	40,000 50,000
Intra/Inter-cluster h-edge cardinality	5 5

How many eigenvectors should we calculate?

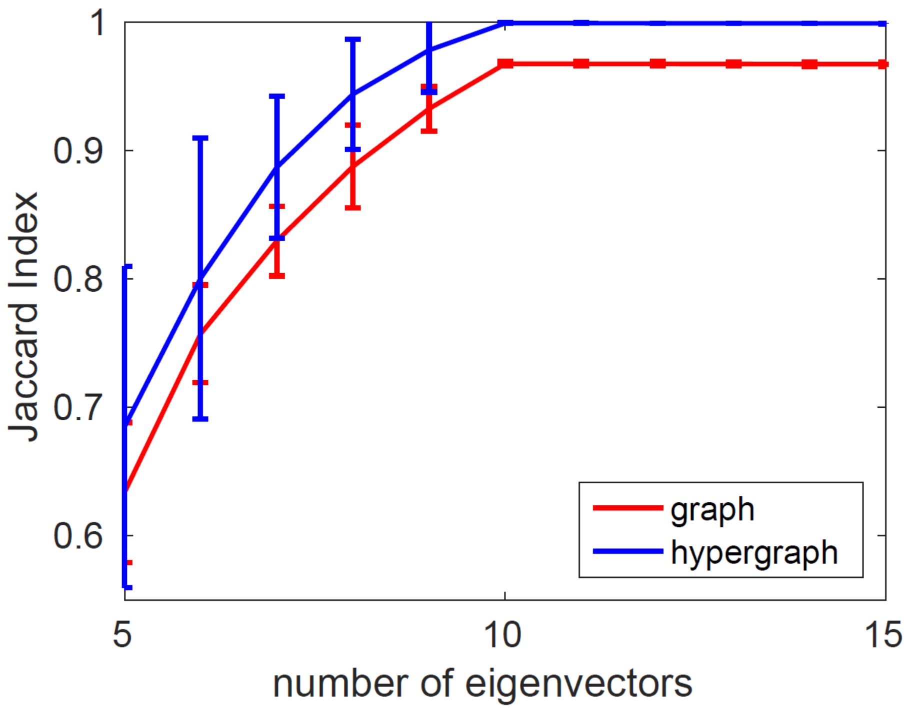


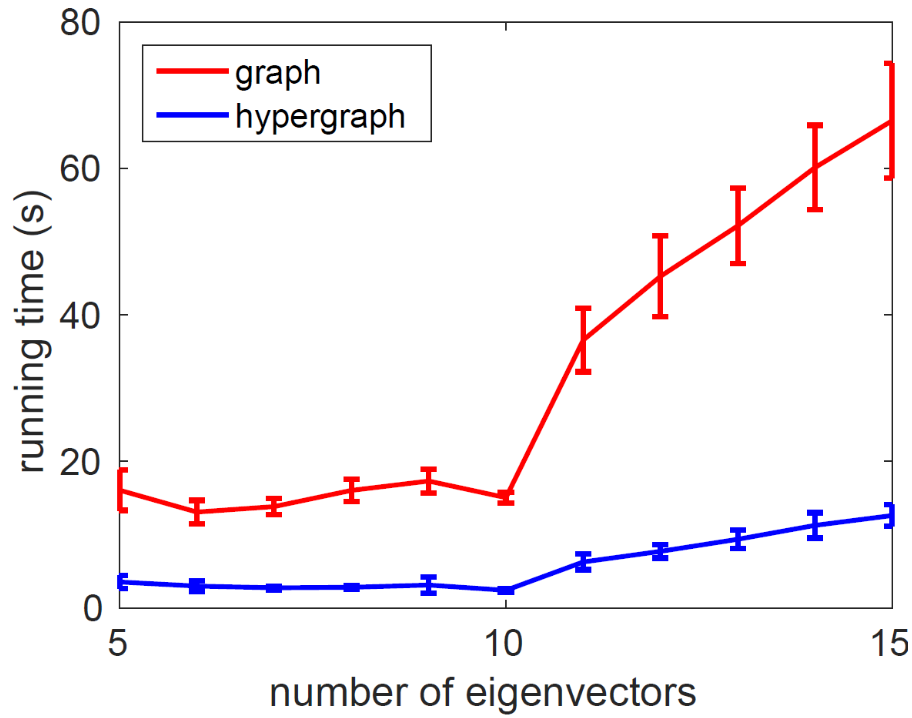
Less noisy data: G1

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	40,000 50,000
Intra/Inter-cluster h-edge cardinality	5 5

How many eigenvectors should we calculate?

Sandia National Laboratories

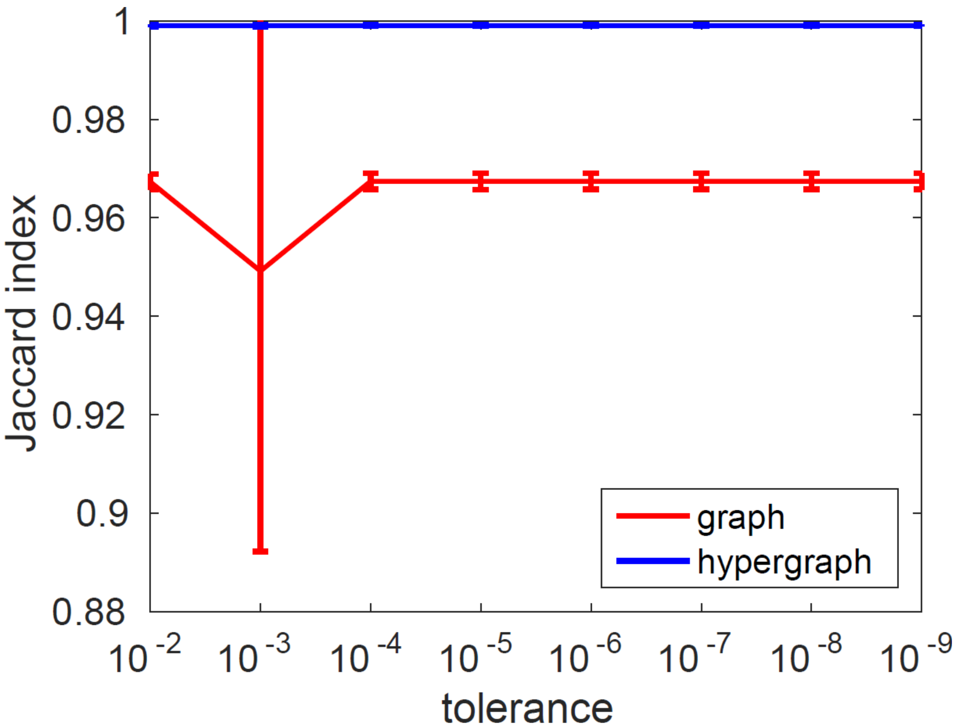


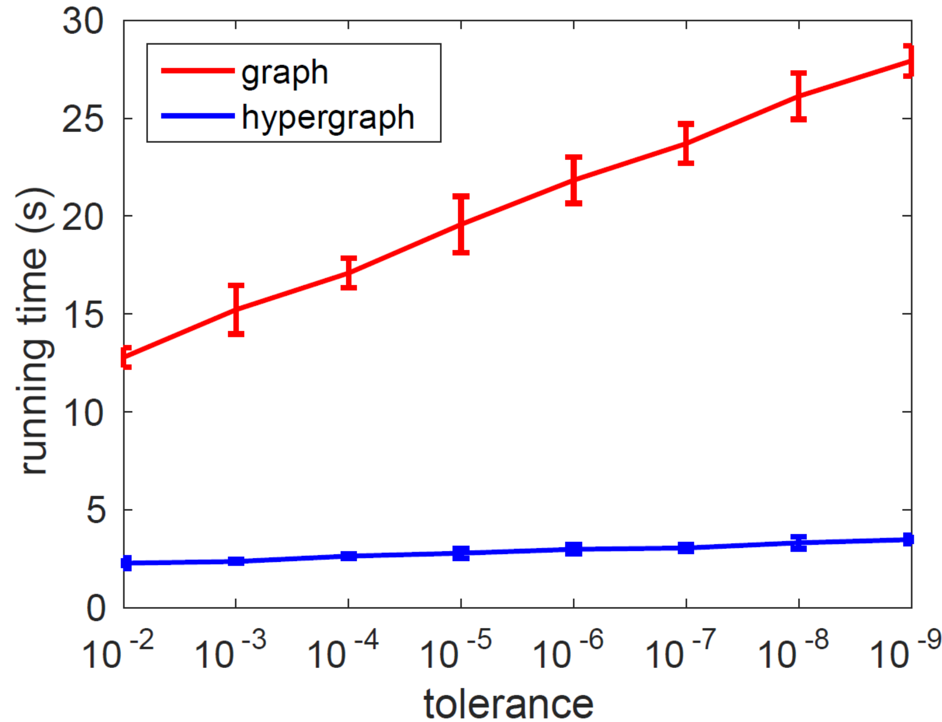


Noisier data: G3

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	5 5

What tolerance should we use?





G3

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	5 5

eigenvectors for tolerance of 10^{-1} were random

Conclusions

- Graph vs hypergraph
  - Preliminary results suggest a **dramatic runtime difference** between eigensolver computation for graph and hypergraph case
  - **Larger Jaccard indices** for hypergraph over graph for several problem classes
- Eigensolver
  - Low tolerances are acceptable
  - Choice of number of eigenvectors is very important
  - LOBPCG is effective for problems we studied
- Currently exploring real world problems where hypergraphs may be a better choice